ISSN No:-2456-2165

Design and Implementation of a Smoke/Fire Detection using Computer Vision and Edge Computing

Nwosu Ifeoma .L.[1]; Alagbu Ekene E.[2*]; Ezeagwu Christopher O.[3]; Okafor Chukwunenye S.[4]; Obiadi Ifeanyi P.[5]
[1,2,3,4]Department of Electronic and Computer Engineering Nnamdi Azikiwe University Awka Anambra State Nigeria

Corresponding Author:- Alagbu Ekene E.[2*]

Abstract:- In Nigeria, over 2,000 fire outbreaks reported resulted in ₦1 trillion worth of property damages. Likewise, a record of 31 market fires was reported in various states in the country. These numbers point to the dangers posed by fires and the need of a faster detection approach. This paper presents a computer vision-based smoke and fire detection system that is run on an edge server. The proposed system consists of a custom convolutional neural network (CNN) model which is utilized to extract features in image frames to identify fire and smoke occurrences. The k-fold cross validation algorithm has been proved on a simplified CNN model which has a small number of layers in order to improve the performance of the image classification. The experimental analysis of the model shows that the proposed system is capable of classifying fire images accordingly with an ROC value of over 0.67 in each class. This model is recommended for use in deep learning tasks that require automatic feature extraction and object detection in image processing applications.

I. INTRODUCTION

Fire is one of the most devastating natural disasters, causing significant property damage and loss of lives each year. Although fires serve as a benefit for human activities such as cooking, when left untended, it poses a danger that also threatens people's livelihood. According to available statistics by Abeku et al (2021), in Nigeria, 31 market fires was documented within a period of 18 months between 2020 - 2021. Likewise, Daily Trust (2022) reported a total of 53 fire cases (in both indoor and outdoor environments) between January to March 2022 in which 19 persons lost their lives. It results in significant property damages such as the ₦1 trillion worth of damage in Nigeria in 2022 (Vanguard, 2023) and $1 billion damages as a result of manufacturing and industrial fires in the US (NFPA, 2019). These fire occurrences are often attributed to faulty electrical wiring, storage of combustive fuels in an indoor environment and improper disposal of cigarette stubs. It could also be as a result of dry weather or an intentional act (arson), but no matter the cause it is important to curb fire outbreaks in time because they are proving a constant threat to people's safety, their possessions, and the environment.

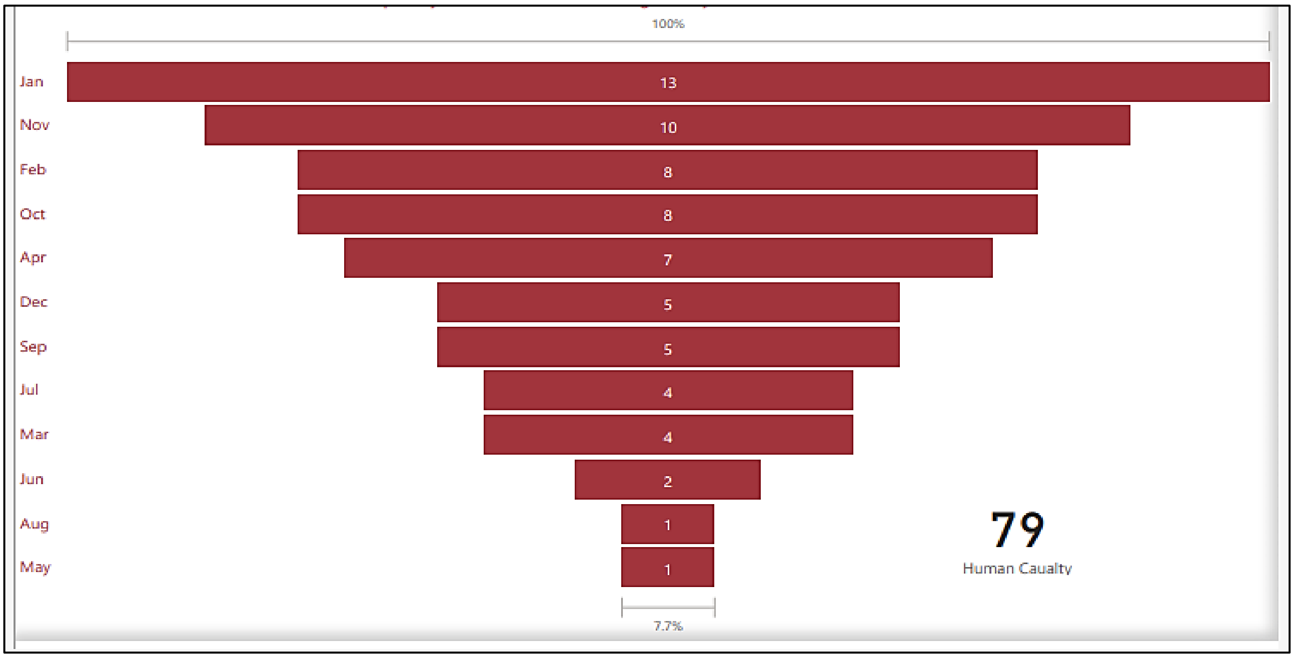


Fig 1 Market Fire Outbreak Frequency in Nigeria within 2019 and 2020 (Orodata, 2020)

Traditional fire detection approaches, fire alarms and smoke sensors, serve the purpose of alerting building residents to imminent fire dangers once the fire or smoke has reached a certain threshold value. This quality makes the use of these devices not well suited for environments where the fire might go unnoticed for a while and still cause significant destruction. Thus, they are not effective to serve as early warning systems for hazardous environments and remote areas (Avazov et al, 2023). These systems also have other limitations like false positives and device malfunction, and these restrictions have the potential to severely impair their efficiency, which could have disastrous effects in emergency situations.

In recent years, advances in computer vision research have prompted the use of surveillance camera data for automatic detection and object recognition. Computer vision, a sub-field of artificial intelligence, allows computers to extract information from images and videos through the use of machine learning algorithms. This has proven to have a versatile application for scenarios such as accident and disaster detection in airports, industries and other open environments, traffic monitoring and control in smart cities and home security (Alamgir, 2020). The limitations of conventional fire detection systems can be effectively addressed by utilizing computer vision. The surveillance cameras, which are already mounted for security purposes, can be used to monitor the surrounding environments for fire occurrences. It also offers an additional advantage because it can be used to verify the intensity and nature of an alarm, if an alarm is false or real, without requiring a physical presence. Edge computing is a computing paradigm that brings processing and data storage closer to the devices where data is generated (Liang et al, 2022). The limitations of conventional fire detection systems can be effectively addressed by the combination of computer vision and edge computing technologies. By deploying computer vision algorithms at the edge, visual data can be analysed quickly. This enhances the system's ability to perform analysis instantly without depending on remote processing centres.

This research proposes the use of these technologies to develop a smoke and fire detection system with improved detection capabilities. The proposed system consists of a custom CNN model which is utilized to extract features in image frames to identify smoke and fire occurrences.

- *Problem Statement*

The problem to be addressed through this study is the issue of the traditional fire detectors in urban environments. These devices, which are commonly in use, suffer from a few disadvantages. Firstly, they have less efficiency in high ceiling buildings and open areas such as factories, markets and warehouses. The buildup of the flames and smoke has to be directly beneath the detector and at a certain level in order to trigger an alarm or response signal. Also, they require consistent maintenance to ensure reliability.

These problems when not properly put in check can lead to great damage of properties and loss of lives. In most recent times, the occurrence of market fires has led also to the collapse of various business, emotional trauma, litigation damage of a brand reputation and financial loss. Smoke, a product of fire, is also known to cause instantaneous loss of lives. Long-term exposure to smoke and other fumes produced by fires can cause several health issues such as breathing difficulties and eye irritation.

Thus, it is of great importance to address these issues as it would mitigate the incurred losses and also lead to the development of a more reliable and efficient fire detection approach.

- *Aim*

The aim of this work is to develop and implement a smoke/fire detection system using computer vision and edge computing.

II. LITERATURE REVIEW

Recently, various approaches proposed by researchers have explored the use of machine learning and deep learning algorithms in the detection of fire and smoke. They explored the use of CNNs and its variants (Kukuk and Kilimci, 2021) and the use of object detection models such as YOLO, SSD and Faster R-CNN (Zheng et al, 2022) in the identification of wildfires occurrences and their spread. The difficulties in algorithm research for vision-based methods can be attributed to the atypical nature fire flame and smoke (Lee and Shim, 2019).

The research by Sheng et al (2021) proposed a statistic image feature-based deep belief network (DBN) for fire detections. DBN was utilized could automatically learn fault features in multiple fire stages. layer by layer using restricted Boltzmann machine (RBM). Yavuz Selim et al (2021), in their paper, utilized transfer learning method using the Inception V3, SequeezeNet, VGG16 and VGG19. Their detection process was split into 3 stages: first, the flame region extraction using basic image processing algorithms. Next, mobility of the flame is analysed by comparing the video frames of the fire image. Afterward the training of the CNN, their model showed a 98.8% classification success on the Inception V3 architecture. Another research examined the development of an energy-efficient system based on CNN for early smoke detection in both normal and foggy IoT environments (Khan et al, 2019). Ren et al (2021) proposed an intelligent detection technology, using fuzzy logic reasoning, for electric fires based on multi-information fusion for green buildings. An implementation of the machine learning wildfire detection and alert system by Ranadive et al (2022) is currently being used in the USA.

On the basis of object detection models, Mukhiddinov, Abdusalomov, and Cho (2022) presented a vision-based fire detection and notification system that utilized smart glasses and deep learning models for blind and visually impaired (BVI) people. Their model employed an improved YOLOv4 model with a convolutional block attention module. Another study by Xue et al (2022) proposed an improved forest fire small-target detection model based on YOLOv5 architecture. Their model showed an improved performance with the mAP metric increasing by over 10.1%. The study by Saponara, Elhanashi, and Gagliardi (2021) presented a real-time video-based fire and smoke detection using YOLOv2 Convolutional Neural Network (CNN) in antifire surveillance systems. Their work was deployed in a low-cost embedded device (Jetson Nano), which was composed of a single, fixed camera per scene, working in the visible spectral range. Xu, H. et al (2022) presented Light-YOLOv5 which modified the YOLOv5 architecture by altering some modules in the network and introducing a global attention mechanism (GAM) for effective feature extraction.

Several studies utilized the ensemble learning approach in order to improve the detection accuracy of the CNN models being adopted. An example of such system can be found in the research by Xu, R. et al (2021). Their work utilized two individual learners (YOLOv5 and EfficientDet) for the detection process and another learner (EfficientNet) for learning global information in order to minimize false positive detection. This resulted in a decrease of false positives by 51.3%. The study by Almoussawi et al proposed a CNN-AE based pipeline for classification and verification of fire-related images. Nguyen et al (2021) proposed a CNN-LSTM network for fire detection and Grari et al (2022) implemented a regression ensemble learning model for predicting fires utilizing NASA's FIRMS dataset.

Based on the literature, the research in smoke and fire detection still requires improvements in the design and implementation of a simplified deep learning model to efficient detection and the provision of a diverse dataset for detection, Likewise, there are a very few studies that focus on monitoring ongoing flames and smoke in near real-time using deep learning methods. As a consequence, we want to continue our study in this field and enhance our findings. This study proposes the application of the k-fold cross validation CNN model to rapidly identify fire occurrences with a low rate of false positives. It further explores the use of the CNN and OpenCV real-time computer vision algorithm to avert numerous fire outbreaks which makes the research novel for study.

III. MATERIALS AND METHODS

This section explores the details of the custom CNN model created for the purpose of the smoke and fire detection system. The Python Keras library was used in the creation of a sequential CNN model utilized for the feature extraction and classification process. A custom dataset was curated for training and testing of the model. The data was gathered from Internet related image tags that related to fire in buildings and other structures in an urban setting.

➢ *Image Preprocessing*

It involves image resizing (adjusting the height and width of the image). The functions from Python OpenCV's library were used in this task to resize the test images to a size of 150 x 150 pixels. Then the images are normalized using equation (1). This process adjusts the brightness of the image and ensures that the image pixels are in [0, 1] range.

$$img_p = \frac{img}{255} \tag{1}$$

➢ *CNN Mathematical Model*

This section of the CNN is in charge of extracting the features in each image for easier detection across the layers. It comprises of the convolutional layer and the pooling layer.

- *Convolutional Layers*

Colored images of $150 \times 150 \times 3$ pixels are fed as the input for the model. The input image is subjected to 16 filters with a size of 3×3, producing 16 feature maps in the first convolution layer. The features are extracted during the convolution process using the filter. The feature map, F is represented by the convolution operation between the input image, M and the filter, T as shown in equation 3.2.

$$F[i,j] = (M * T)_{[i,j]} \tag{2}$$

The ij-th entry of the feature maps is as shown in equation 3.3

$$f[i,j] = \sum_x^{h_m} \sum_y^{w_m} \sum_z^{n} M_{[x,y,z]}\, T_{[i+x-1,\ j+y-1,\ z]} \tag{3}$$

- *Pooling Layers*

In the max pooling layer, a pool size of 2×2 pixels is used to choose the maximum activations of these 16 feature maps with a stride of 3×3 pixels. The stride indicates how far the pooling matrix moves for each pooling step; this results in a reduction in the size of the feature maps. The pooling layer ensures that the most relevant details (the maximum values) are kept while removing the less significant ones (the minimum values).

$$P = \phi_p(M * T) \tag{4}$$

Where ϕ_p is the pooling function. The dimension of the pooling layer is gotten from the formula in equation 3.5, where $h_m \times w_m$ represents the dimensions of the input, $h_t \times w_t$ represents the dimensions of the filter, s is the stride length and n is the number of channels in the input.

$$\dim of\ P = \left(\frac{h_m+1-h_t}{s}\right) \times \left(\frac{w_m+1-w_t}{s}\right) \times n \tag{5}$$

- *Flattening Layer*

The classification process of the CNN starts off with a flattening layer which reduces its input (the stacked output of the convolutional and pooling layers) into a 1-dimensional shape for ease of computation. Its output is passed to the Dense layers.

- *Dense Layer*

The first dense layer has 32 classes and the second one (the last layer of the CNN) has four classes (i.e., the classification in the dataset). The mathematical operation of the dense layer is as given in equation 3.6 where ϕ_d is the activation function of the dense layer, P is the input to the layer, w is the kernel and b is the bias of the layer.

$$z = \phi_d(\sum_i w_i P_i + b) \quad (6)$$

- *Activation Functions*

The ReLU activation function is used for all the layers except the last dense layer which uses a Softmax activation function. The ReLU function in this work is the standard function which selects an element-wise maximum of 0 and the input data. It introduces a bias value, b on the convoluted output of each layer and it is calculated as seen in Equation 3.7, where c is the output at the layer where the activation function is applied.

$$c = \phi_a(M * T + b) \quad (7)$$

The Softmax function, used in the last dense layer, acts as a classifier because it returns a vector as a probability value and the elements of its output vector results as 1. In this case, we have four probability classifications for the model. The Softmax activation is performed using equation 3.9 where z is the output of the previous dense layer.

$$softmax(z) = \frac{\exp(z)}{\sum \exp(z)} \quad (8)$$

- *Algorithm and Flowchart*

The model is built with the following algorithmic steps:

- *Import necessary libraries and modules: This includes Keras, sklearn, os, numpy, and cv2.*
- *Initialize necessary lists: This includes a list to store models, histories, test predictions, and test labels.*
- *Load and pre-process the data:*
  - ✓ *Get the list of all images and their corresponding labels from the data directory.*
  - ✓ *Convert labels to integers and then one-hot encode them.*
  - ✓ *Convert one-hot encoded labels to single labels.*
- *Perform k-fold Cross-Validation:*
  - ✓ *For each fold in the k-fold Cross-Validation:*
    - *Initialize a Sequential model and add layers to it. This includes Conv2D, MaxPooling2D, Flatten, Dense, and Dropout layers.*
    - *Compile the model with Adam optimizer, categorical cross-entropy loss, and accuracy metrics.*
    - *Load and pre-process the training and test images.*
    - *Fit the model on the training data and validate it on the test data.*
    - *Use the model to predict the test set and store the predictions and actual labels in the lists.*
    - *Save the model in the list of models.*
- *Calculate the average accuracy and loss at each epoch.*

These implementation steps serve to ensure that the CNN is trained on a diverse set of data and that its operation is evaluated on key metrics such as the loss and accuracy. A diagrammatic flow of the design process detailing the interconnections of the model design can be seen in Figure 2.

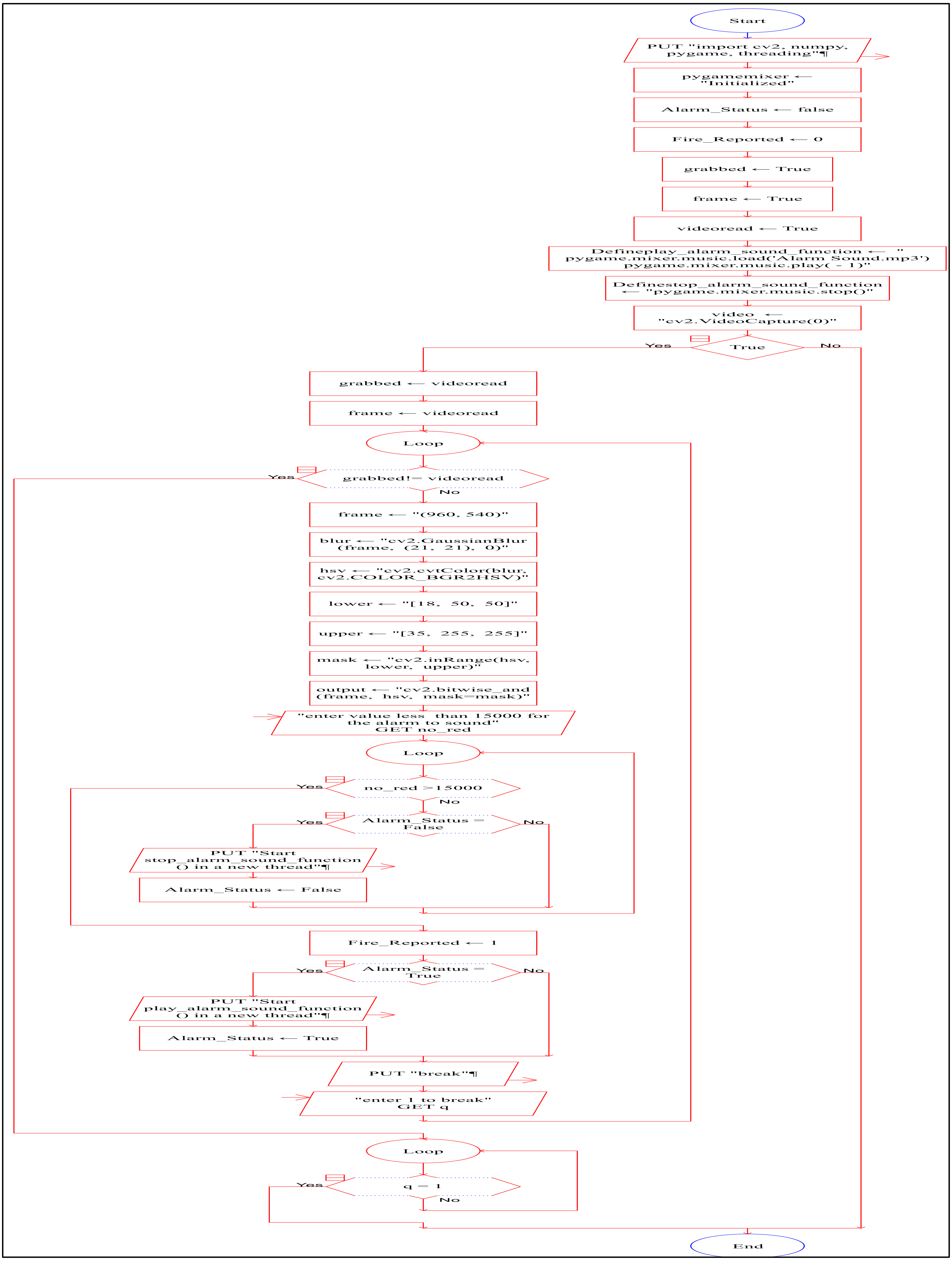


Fig 2 Flowchart

IV. RESULT AND ANALYSIS

This section provides a detailed overview of the performance of the k-fold cross-validation CNN model on the image dataset. It is evaluated on the performance of key metrics of a deep learning model. The CNN model was implemented using Python TensorFlow and Keras platforms. Training and validation were performed on a Lenovo PC with Windows 11 Pro 64-bit OS, 11th Gen Intel® Core i5 Processor with a frequency rate of 2.40GHz and 8GB RAM size.

The model is a multi-class classification algorithm that classifies into one of four categories; thus a 4 x 4 confusion matrix is generated which shows the number of actual labels against the number of predicted labels in each class (Figure 3). The images are classified into any of the four classes namely smoke (class 0), fire (class 1), fire and smoke (class 2) and neither (class 3). The diagonal of the matrix represents the number of correctly predicted images in each class. The overlapped region between classes in the matrix represent the number of misclassified images between the classes and it is worth noting that the model learned the differences between fire/smoke images from normal images with smoke images being correctly classified in 580 images.

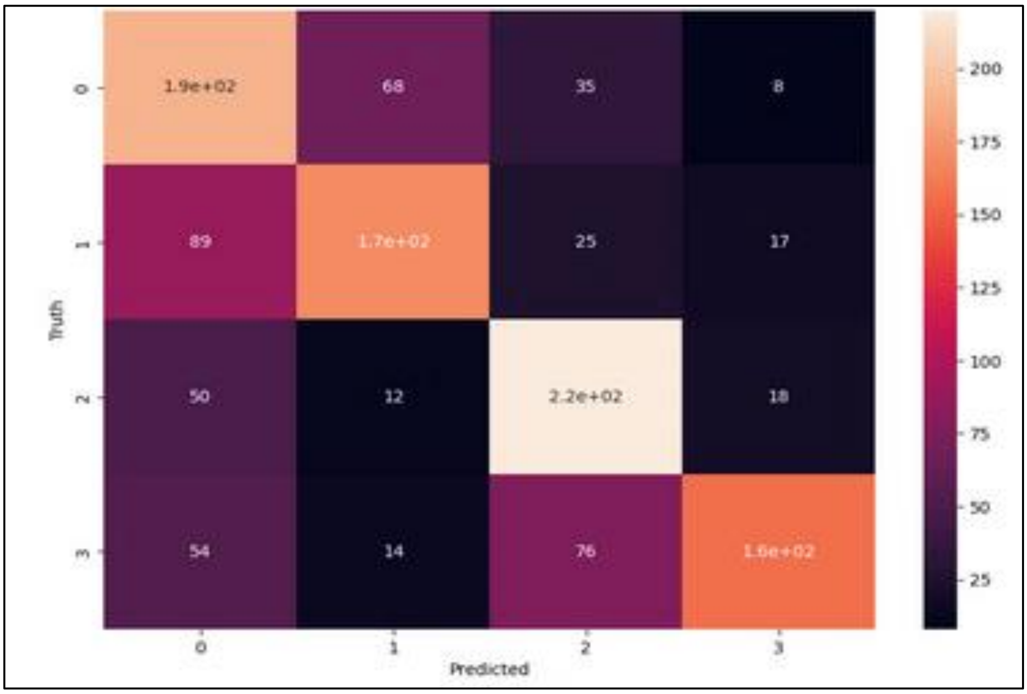
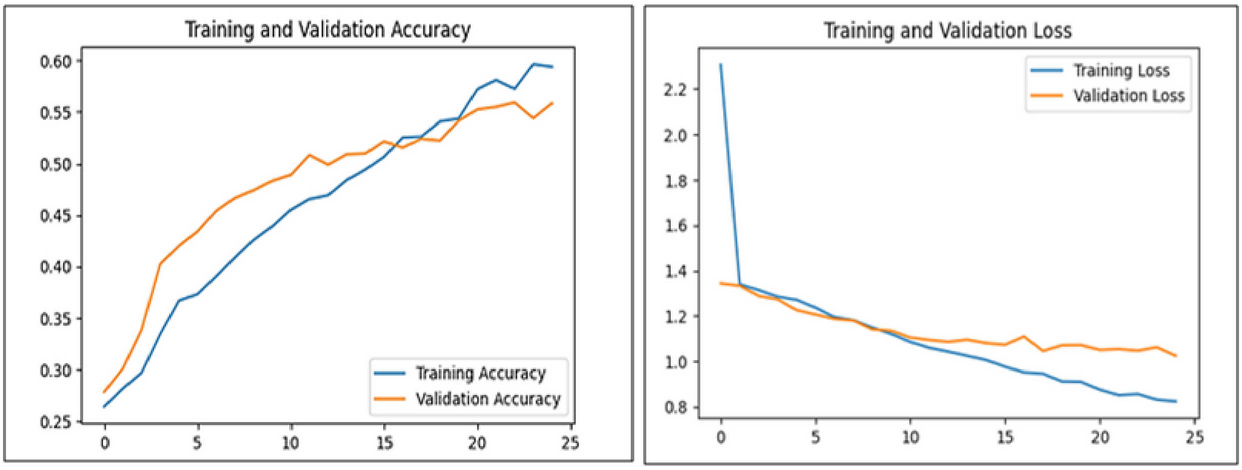


Fig 3 The Confusion Matrix of True and Predicted Labels of the Proposed Model

The accuracy of our model over the training and validation data is evaluated as shown in Figure 4. A total of 25 epochs was used to train the CNN model and a maximum accuracy of over 60% was recorded. The model is then evaluated in terms of the loss metric. The model produces a minimal loss of 0.8 after training due to implementation of thee cross-validation algorithm and the Adam optimizer function.



(a) (b)

Fig 4 (a) Graph Depicting a Gradual Increase in Training and Validation Accuracy (b) Graph Depicting a Steady Reduction for the Training and Validation Loss for the CNN Model

The ROC curves for the model shows the varying AUC values of the classes defined in the code. Thus, the AUC values of 0.67, 0.70, 0.76 and 0.69 correspond to classes 0, 1, 2 and 3 respectively. This implies that the model has a 67% probability of classifying images in class 0, 70 % for class 1, 76% for class 2 and 69% for class 3.

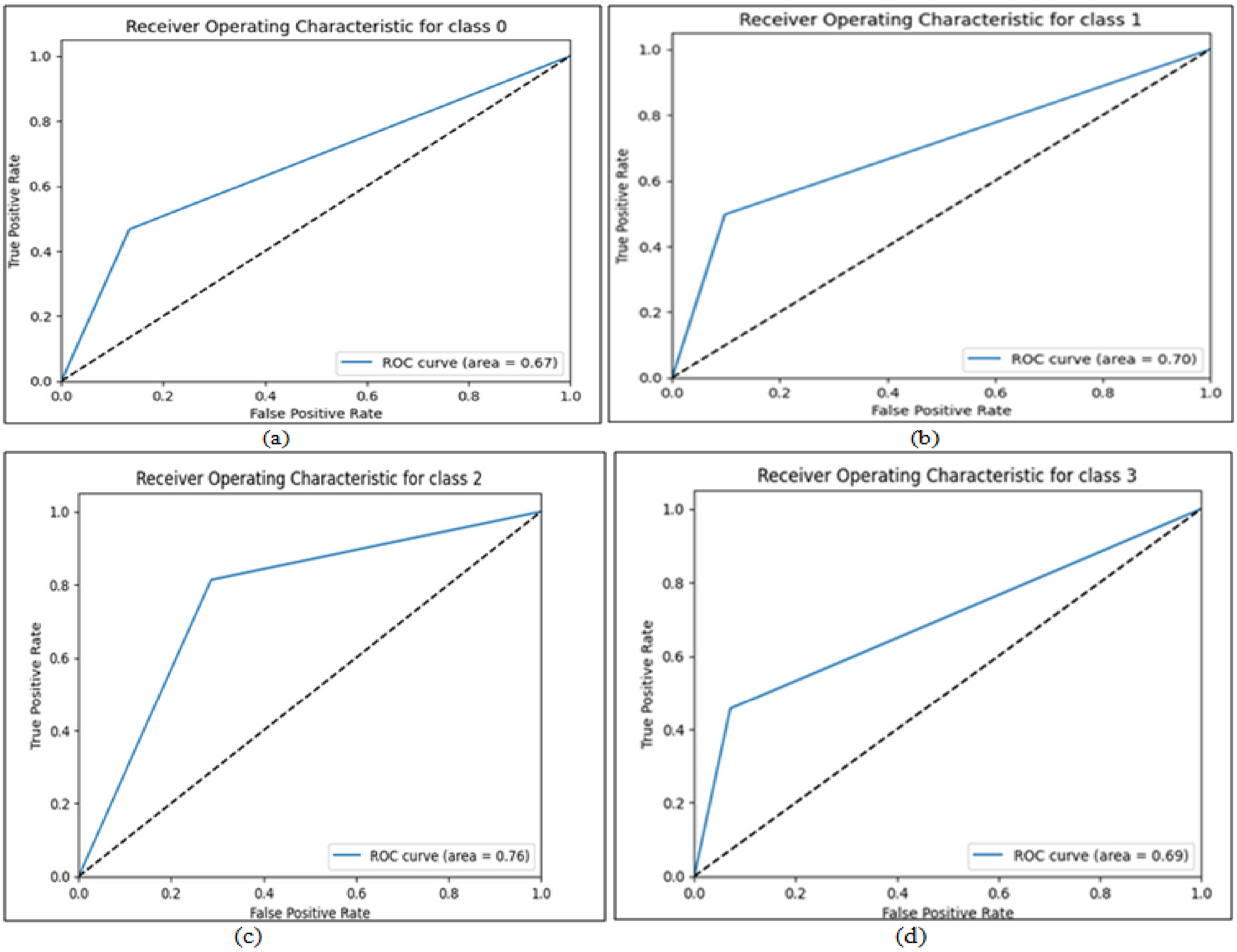
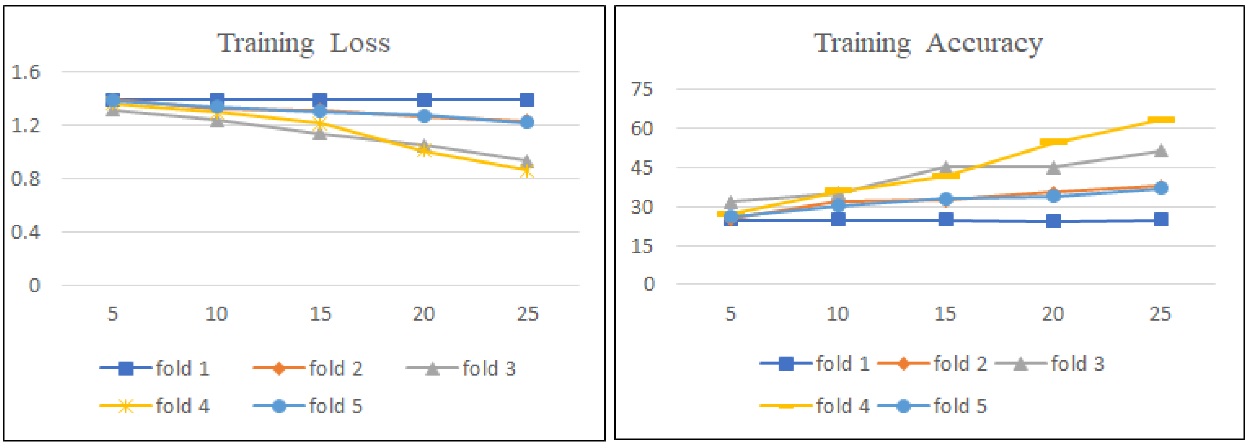


Fig 5 Graph Depicting the ROC Curves of the Proposed Model with Specified AUC Values of each Classification Category as it Relates to (a) Class 0 (b) Class 1 (c) Class 2 (d) Class 3 of the CNN Model

➢ *Performance Analysis of the Cross-Validation Algorithm*

The cross validation is implemented for the model since it has few images for training the neural network. This is achieved through k-fold cross validation with five folds in order to reduce the computational complexity. The performance analysis of the model in terms of the training accuracy and loss for the different folds of the algorithm is depicted in the graphs in Figure 6. It shows a steady improvement in the accuracy and loss metrics across the training epochs and the average of these values were computed to obtain the final performance of the system.



(a) (b)

Fig 6 Accuracy and loss performance of the CNN across the folds of the cross-validation algorithm after

V. DISCUSSION

While there are a lot of false positives and false negatives in the case of smokes, the custom convolutional neural network has a high accuracy rate for detecting flames in both indoor and outdoor contexts. Efficient early smoke detection is another feature of the developed model. To construct a simulated alarm system setting, play sound module of the OpenCV library was utilized. The "Alarm Sound.mp3" mp3 audio file was linked to the deep learning model and access to the webcam is also granted to the program. Once smoke or a potential fire is detected through the captured image frames, the alarm sound is played and the user is alerted to the impending danger.

VI. CONCLUSION

This research presented a novel computer-vision based fire and smoke detection system that is running on an edge server. The proposed system aims to overcome the issues of fire and smoke detection in two stages: First, the captured image is pre-processed so as to highlight the key features present in the image. This was achieved using the OpenCV library implementation in Python programming language. Afterwards, the CNN model is trained on the processed image in order to properly classify it into the related class (fire, smoke and neither). The CNN model was built using Python programming and the deep learning framework, Keras, was utilized to ensure ease of implementation.

The use of a k-fold cross validation algorithm has been proved on a simplified CNN model which has a small number layers in order to improve the performance of the image classification. The experimental analysis of the model show that the proposed system is capable of classifying fire and smoke images accordingly with an ROC value of over 0.67 in each class. Likewise, the accuracy is observed to increase by 15% across each fold of the training process. This model is recommended for use in deep learning tasks that require automatic feature extraction and object detection in image processing applications.

➢ *Future Research on the Topic can be Carried Out in Areas Pertaining to:*

- The investigation of the use of other deep architectures like Deep Belief Networks and Recurrent Neural Network for the detection of smoke and fire.
- The fire detection using a moving camera.
- The monitoring of the fire spread and direction of movement.

REFERENCES

[1]. Abeku, T., Michael, D., Akpan-Nsoh, I., Udeajah, G., Ogugbuaja, C., Oyewole, R., Godwin, A., & Agboluaje, R. (2021). *Losses to market fires hit N41.54 billion in two years.* The Guardian Newspaper. https://guardian.ng/news/losses-to-market-fires-hit-n41-54-billion-in-two-years/

[2]. Alamgir, N. (2020). *Computer Vision Based Smoke and Fire Detection for Outdoor Environments.* Published PhD Dissertation, School of Electrical Engineering and Robotics, Science and Engineering Faculty, Queensland University of Technology. https://eprints.qut.edu.au/201654/1/Nyma_Alamgir_Thesis.pdf

[3]. Avazov, K., Hyun, A. E., Sami S, A. A., Khaitov, A., Abdusalomov, A. B., & Cho, Y. I. (2023). Forest fire detection and notification method based on AI and IoT approaches. *Future Internet*, *15*(2), 61. doi:10.3390/fi15020061

[4]. Daily Trust (2022). *How Fire Wreaked 700 shops, Killed 19 in 3 months.* Daily Trust. https://dailytrust.com/how-fire-wreaked-700-shops-killed-19-in-3-months/

[5]. Grari, M., Idrissi, I., Boukabous, M., Moussaoui, O., Azizi, M., & Moussaoui, M. (2022). Early wildfire detection using machine learning model deployed in the fog/edge layers of IoT. *Indonesian Journal of Electrical Engineering and Computer Science*, *27*(2), 1062. doi:10.11591/ijeecs.v27.i2.pp1062-1073

[6]. Khan, S., Muhammad, K., Mumtaz, S., Baik, S. W., & de Albuquerque, V. H. C. (2019). Energy-efficient deep CNN for smoke detection in foggy IoT environment. *IEEE Internet of Things Journal*, *6*(6), 9237-9245.

[7]. Kukuk, S. B., & Kilimci, Z. H. (2021). Comprehensive analysis of forest fire detection using deep learning models and conventional machine learning algorithms. *International Journal of Computational and Experimental Science and Engineering*, *7*(2), 84–94. doi:10.22399/ijcesen.950045

[8]. Lee, Y., & Shim, J. (2019). False positive decremented research for fire and smoke detection in surveillance camera using spatial and temporal features based on deep learning. *Electronics*, *8*(10), 1167. doi:10.3390/electronics8101167

[9]. Liang, S., Wu, H., Zhen, L., Hua, Q., Garg, S., Kaddoum, G., Hassan, M. M., & Yu, K. (2022). Edge YOLO: Real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, *23*(12), 25345-25360.

[10]. Mukhiddinov, M., Abdusalomov, A. B., & Cho, J. (2022). Automatic fire detection and notification system based on improved YOLOv4 for the blind and visually impaired. *Sensors (Basel, Switzerland)*, *22*(9), 3307. doi:10.3390/s22093307

[11]. NFPA (2019). *Fires in U.S. Industrial or Manufacturing Properties.* https://www.vosslawfirm.com/blog/fire-statistics-for-industrial-and-manufacturing-properties.cfm

[12]. Nguyen, M. D., Vu, H. N., Pham, D. C., Choi, B., & Ro, S. (2021). Multistage real-time fire detection using convolutional neural networks and long short-term memory networks. *IEEE Access: Practical Innovations, Open Solutions*, *9*, 146667–146679. doi:10.1109/access.2021.3122346

[13]. Ranadive, O., Kim, J., Lee, S., Cha, Y., Park, H., Cho, M., & Hwang, Y. K. (2022). Image-based Early Detection System for Wildfires. *arXiv preprint arXiv:2211.01629*.

[14]. Ren, X., Li, C., Ma, X., Chen, F., Wang, H., Sharma, A., Gaba, G. S., & Masud, M. (2021). Design of multi-information fusion based intelligent electrical fire detection system for green buildings. *Sustainability*, *13*(6), 3405. doi:10.3390/su13063405

[15]. Saponara, S., Elhanashi, A., & Gagliardi, A. (2021). Real-time video fire/smoke detection based on CNN in antifire surveillance systems. *Journal of Real-Time Image Processing*, *18*(3), 889–900. doi:10.1007/s11554-020-01044-0

[16]. Sheng, D., Deng, J., Zhang, W., Cai, J., Zhao, W., & Xiang, J. (2021). A Statistical Image Feature-Based Deep Belief Network for Fire Detection, *Complexity*, *vol. 2021*, 1-12, doi: 10.1155/2021/5554316

[17]. Vanguard (2023). *Nigeria recorded 2,056 fire incidents, N1trn losses in 2022 – GOC.* https://www.vanguardngr.com/2023/03/nigeria-recorded-2056-fire-incidents-n1trn-losses-in-2022-goc/

[18]. Xu, H., Li, B., & Zhong, F. (2022). Light-YOLOv5: A lightweight algorithm for improved YOLOv5 in complex fire scenarios. *Applied Sciences*, *12*(23), 12312.

[19]. Xu, R., Lin, H., Lu, K., Cao, L., & Liu, Y. (2021). A forest fire detection system based on ensemble learning. *Forests*, *12*(2), 217. doi:10.3390/f12020217

[20]. Xue, Z., Lin, H., & Wang, F. (2022). A small target forest fire detection model based on YOLOv5 improvement. *Forests*, *13*(8), 1332. doi:10.3390/f13081332

[21]. Yavuz Selim, T., Koklu, M., & Altin, M. (2021). Fire Detection in Images Using Framework Based on Image Processing, Motion Detection and Convolutional Neural Network. *International Journal of Intelligent Systems and Applications in Engineering (IJISAE)*, *9*(4), 171-177.

[22]. Zheng, X., Chen, F., Lou, L., Cheng, P., & Huang, Y. (2022). Real-time detection of full-scale forest fire smoke based on deep convolution neural network. *Remote Sensing*, *14*(3), 536. doi:10.3390/rs14030536